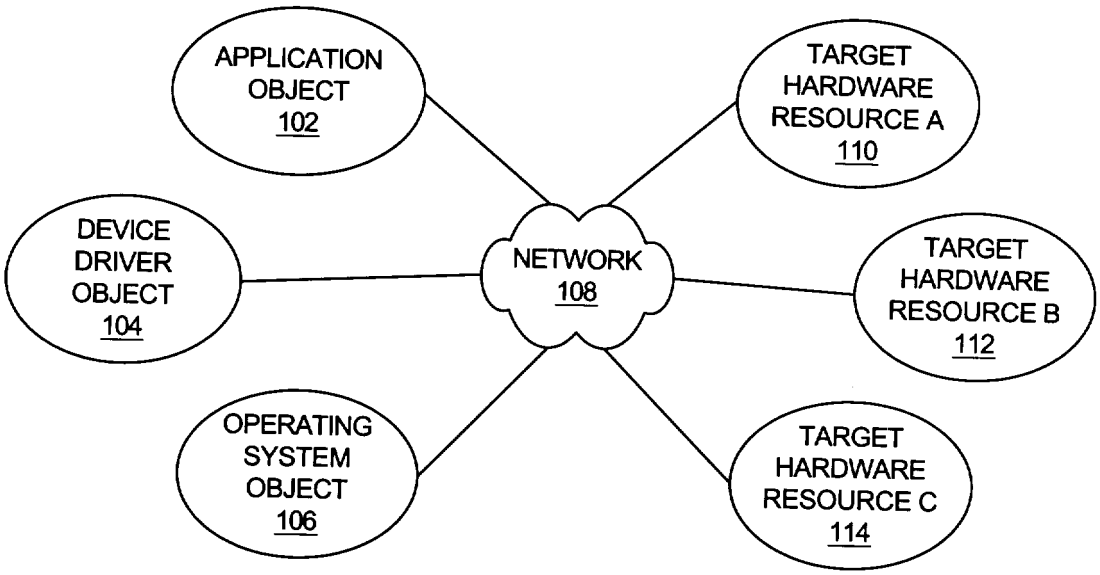

FIG. 1
(PRIOR ART)

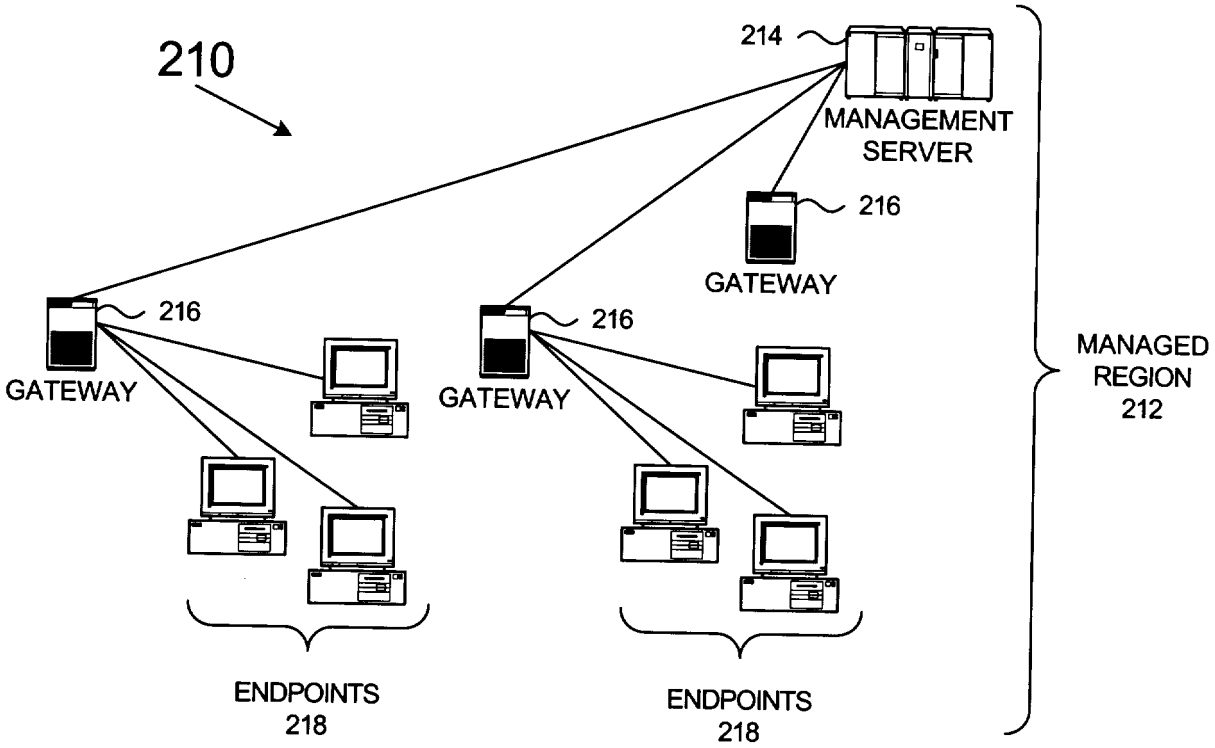

FIG. 2A

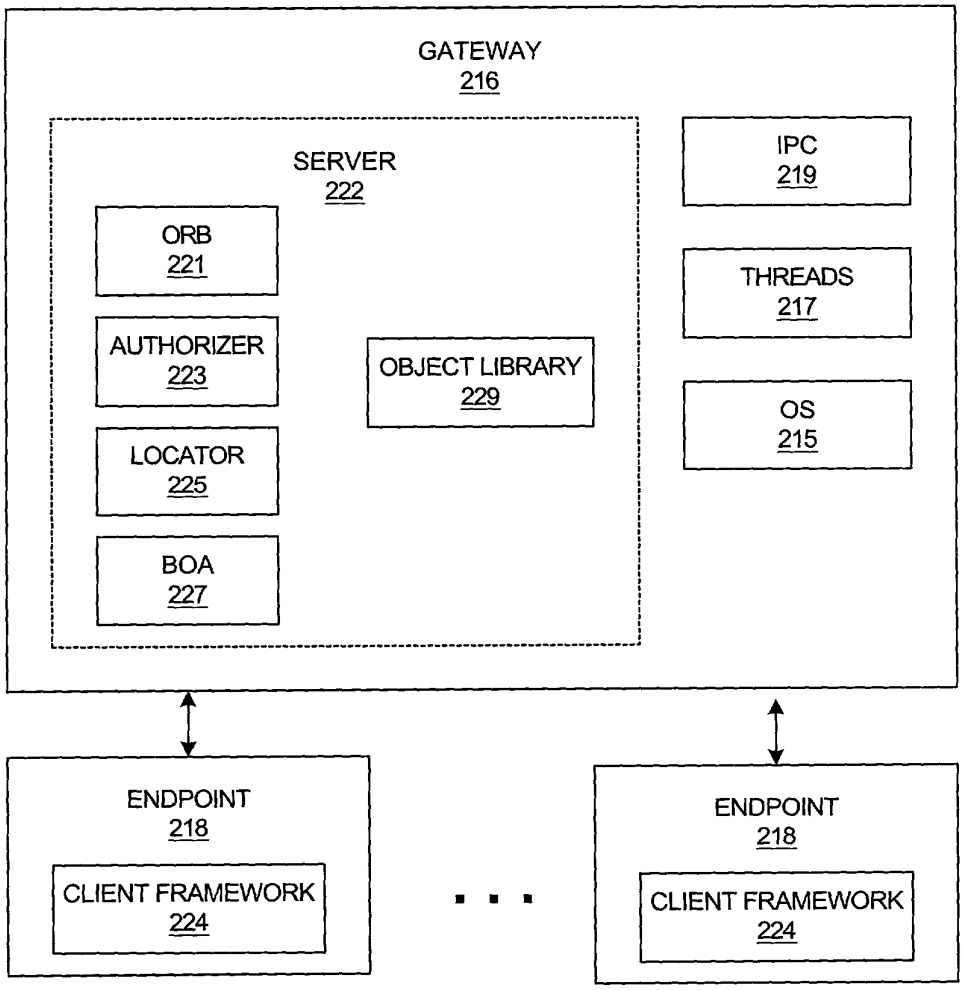
GATEWAY
216
SERVER
222
ORB
221
AUTHORIZER
223
LOCATOR
225
BOA
227
OBJECT LIBRARY
229
IPC
219
THREADS
217
OS
215
ENDPOINT
218
CLIENT FRAMEWORK
224
ENDPOINT
218
CLIENT FRAMEWORK
224

FIG. 2B

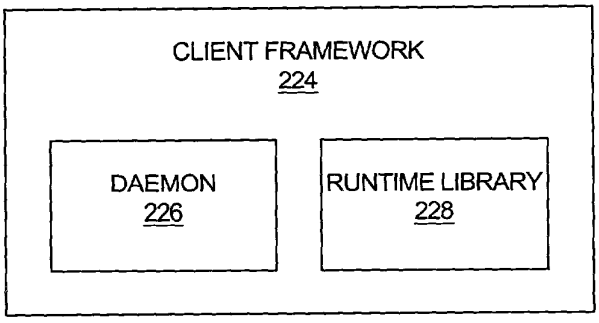
CLIENT FRAMEWORK
224
DAEMON
226
RUNTIME LIBRARY
228

FIG. 2C

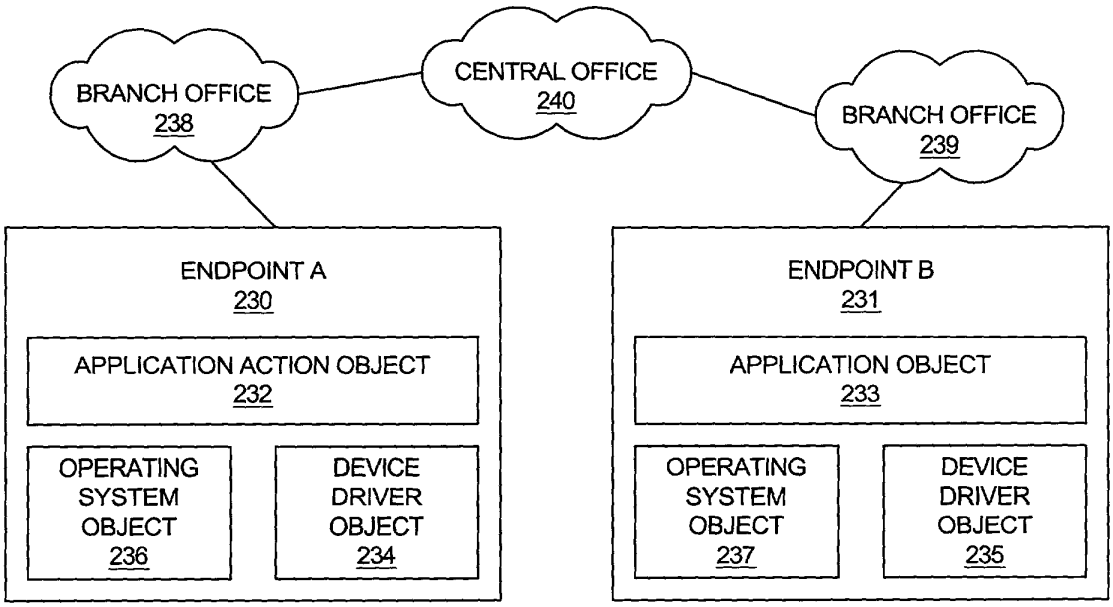
BRANCH OFFICE
238
CENTRAL OFFICE
240
BRANCH OFFICE
239
ENDPOINT A
230
APPLICATION ACTION OBJECT
232
OPERATING SYSTEM OBJECT
236
DEVICE DRIVER OBJECT
234
ENDPOINT B
231
APPLICATION OBJECT
233
OPERATING SYSTEM OBJECT
237
DEVICE DRIVER OBJECT
235

FIG. 2D

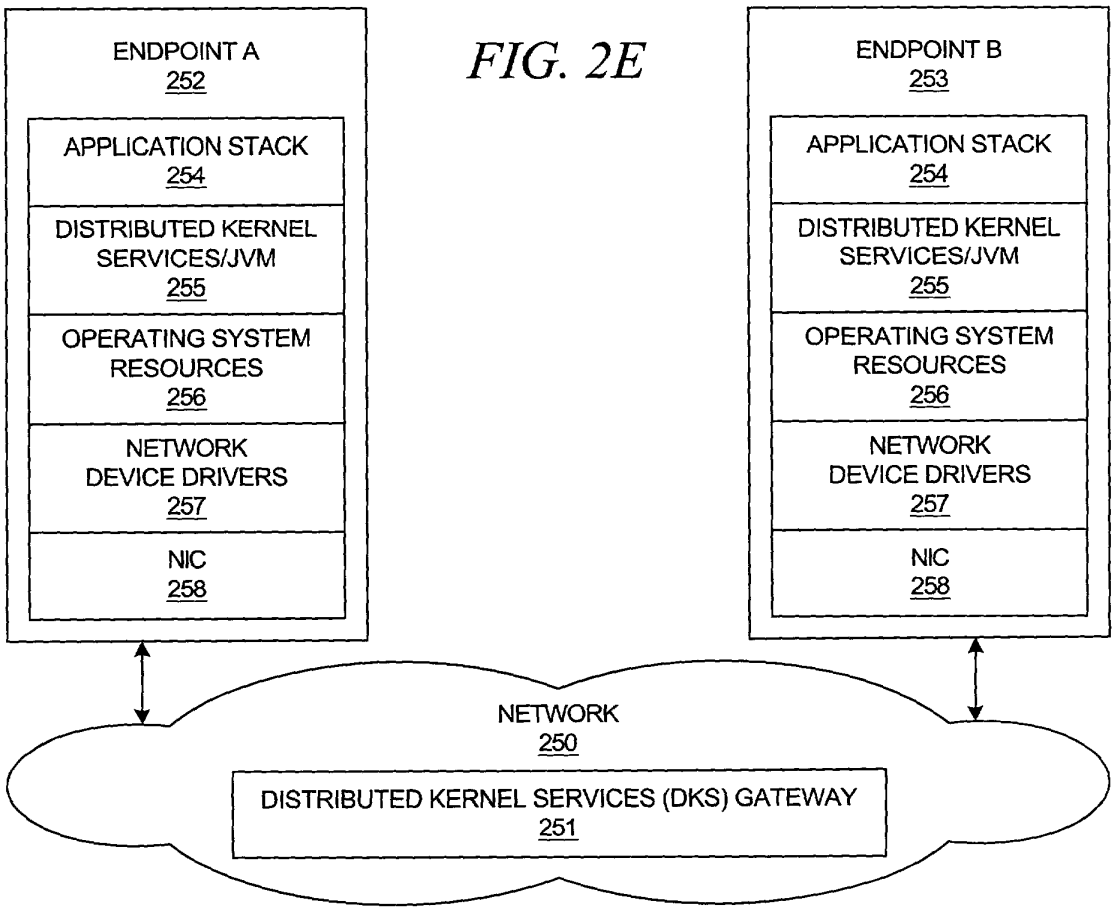
ENDPOINT A
252
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
ENDPOINT B
253
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
NETWORK
250
DISTRIBUTED KERNEL SERVICES (DKS) GATEWAY
251

FIG. 2E

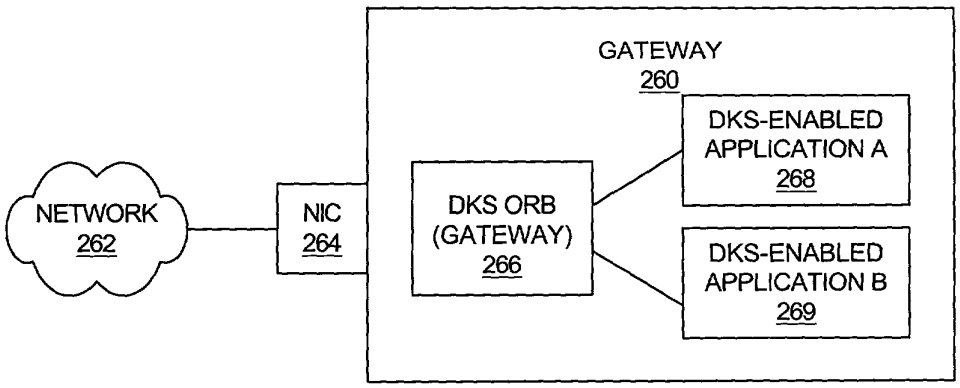
GATEWAY
260
DKS-ENABLED APPLICATION A
268
NETWORK
262
NIC
264
DKS ORB (GATEWAY)
266
DKS-ENABLED APPLICATION B
269

FIG. 2F

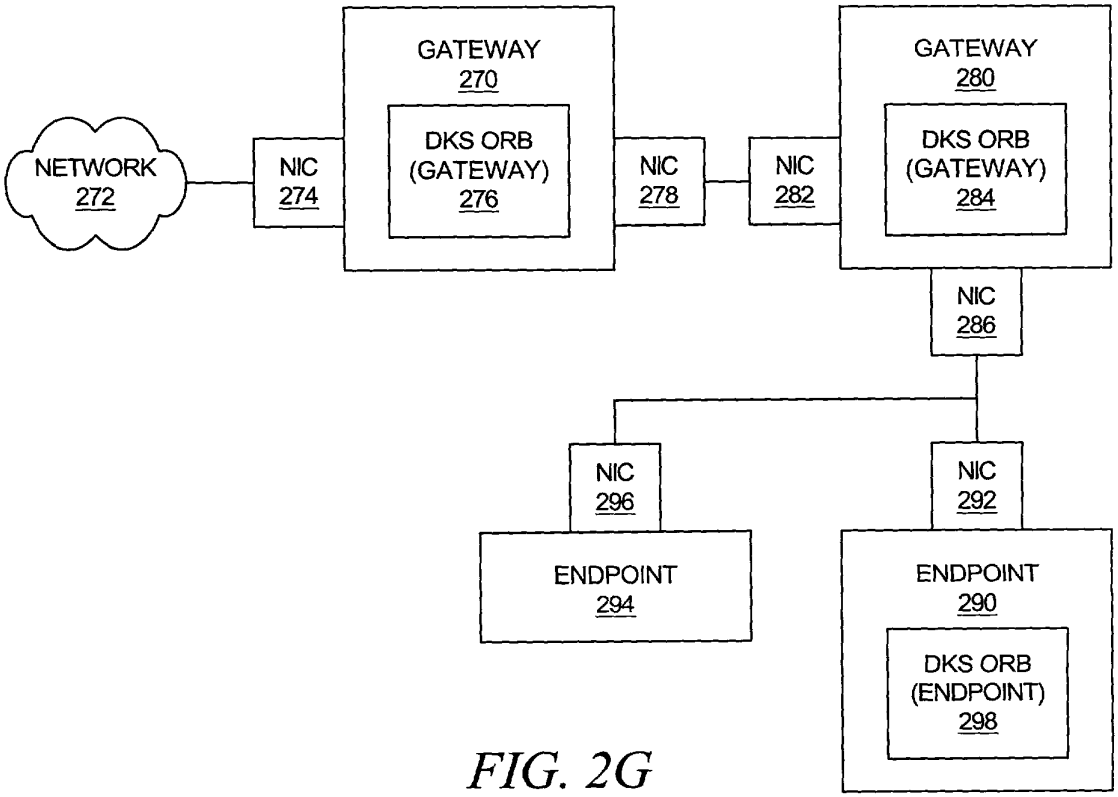
GATEWAY
270
GATEWAY
280
NETWORK
272
NIC
274
DKS ORB (GATEWAY)
276
NIC
278
NIC
282
DKS ORB (GATEWAY)
284
NIC
286
NIC
296
NIC
292
ENDPOINT
294
ENDPOINT
290
DKS ORB (ENDPOINT)
298

FIG. 2G

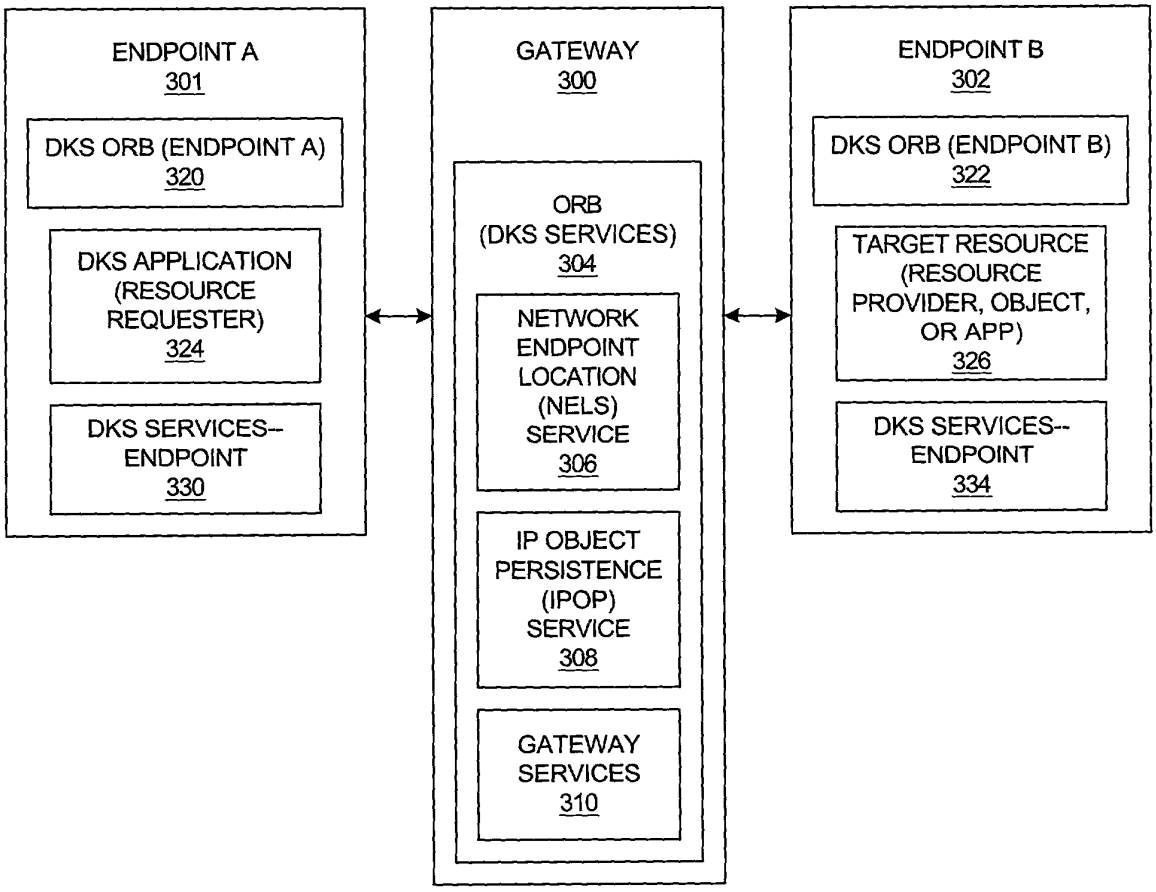
ENDPOINT A
301
DKS ORB (ENDPOINT A)
320
DKS APPLICATION (RESOURCE REQUESTER)
324
DKS SERVICES--ENDPOINT
330
GATEWAY
300
ORB (DKS SERVICES)
304
NETWORK ENDPOINT LOCATION (NELS) SERVICE
306
IP OBJECT PERSISTENCE (IPOP) SERVICE
308
GATEWAY SERVICES
310
ENDPOINT B
302
DKS ORB (ENDPOINT B)
322
TARGET RESOURCE (RESOURCE PROVIDER, OBJECT, OR APP)
326
DKS SERVICES--ENDPOINT
334

FIG. 3

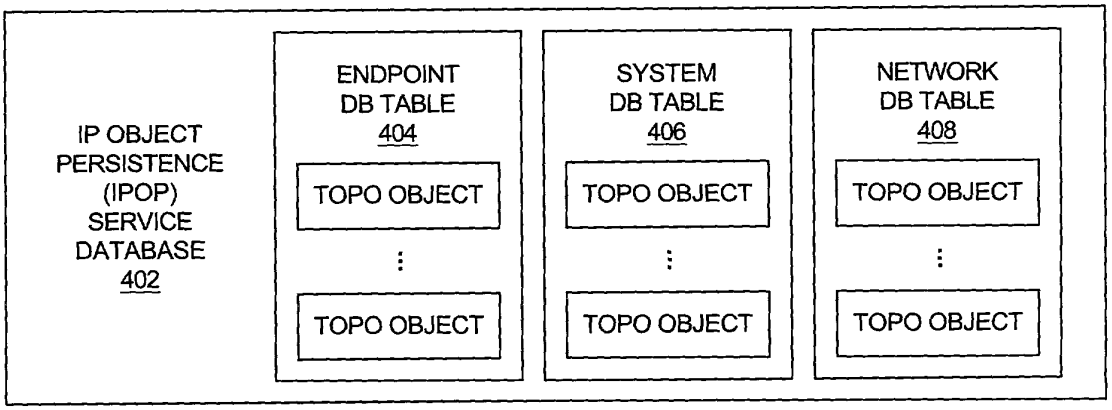
IP OBJECT PERSISTENCE (IPOP) SERVICE DATABASE
402
ENDPOINT DB TABLE
404
TOPO OBJECT
TOPO OBJECT
SYSTEM DB TABLE
406
TOPO OBJECT
TOPO OBJECT
NETWORK DB TABLE
408
TOPO OBJECT
TOPO OBJECT

FIG. 4

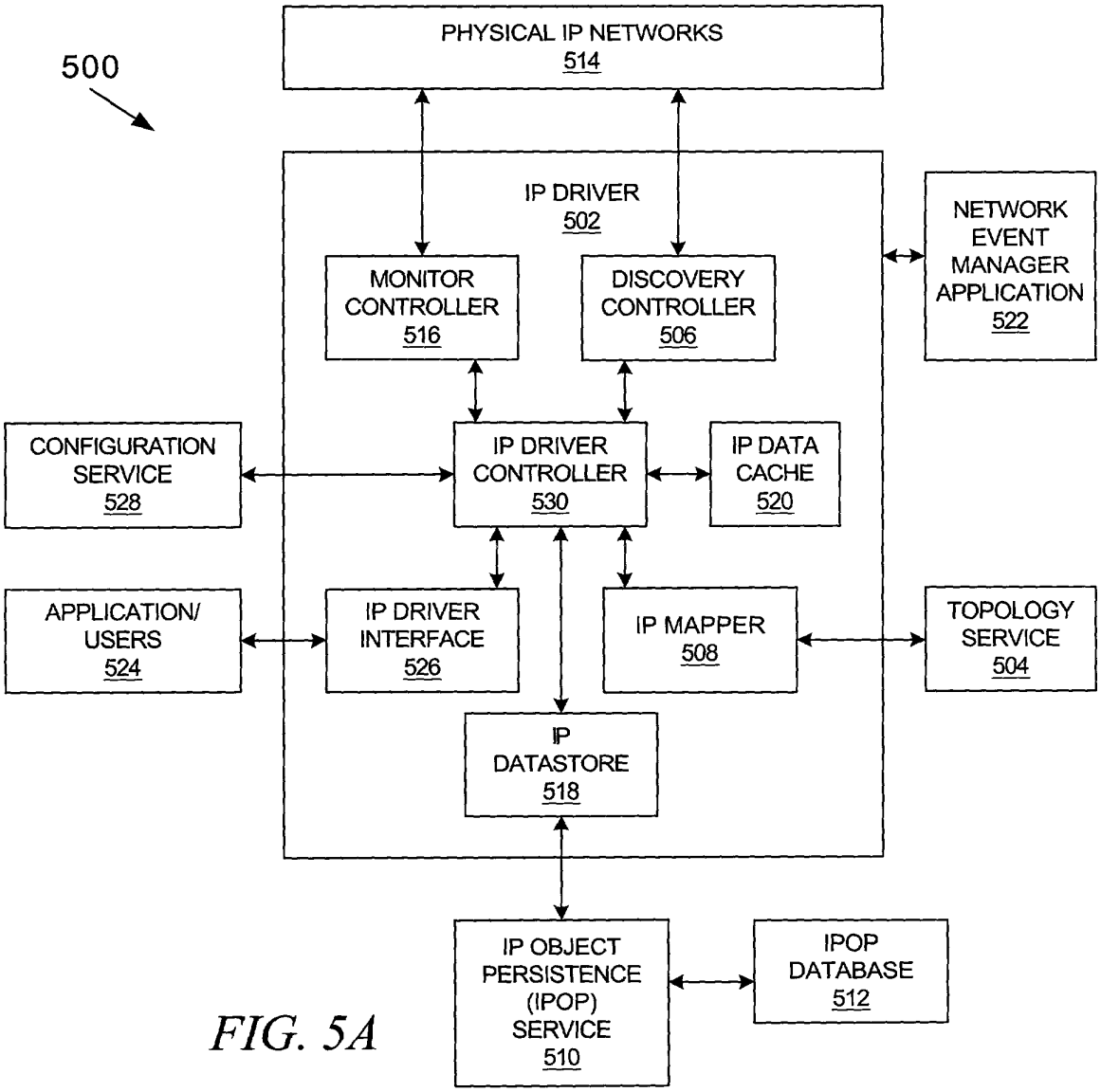
500
PHYSICAL IP NETWORKS
514
IP DRIVER
502
MONITOR CONTROLLER
516
DISCOVERY CONTROLLER
506
NETWORK EVENT MANAGER APPLICATION
522
CONFIGURATION SERVICE
528
IP DRIVER CONTROLLER
530
IP DATA CACHE
520
APPLICATION/ USERS
524
IP DRIVER INTERFACE
526
IP MAPPER
508
TOPOLOGY SERVICE
504
IP DATASTORE
518
IP OBJECT PERSISTENCE (IPOP) SERVICE
510
IPOP DATABASE
512

FIG. 5A

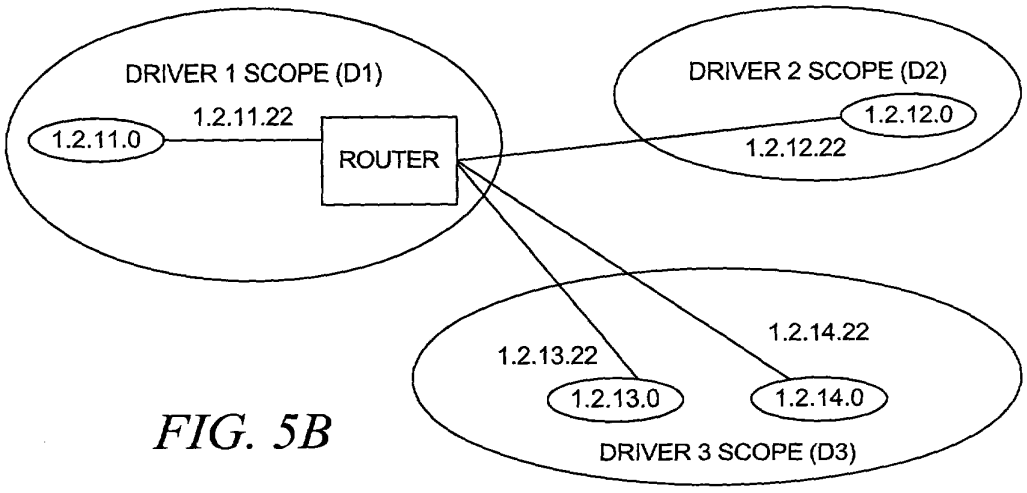
DRIVER 1 SCOPE (D1)
1.2.11.22
1.2.11.0
ROUTER
DRIVER 2 SCOPE (D2)
1.2.12.0
1.2.12.22
1.2.14.22
1.2.13.22
1.2.13.0
1.2.14.0
DRIVER 3 SCOPE (D3)

FIG. 5B

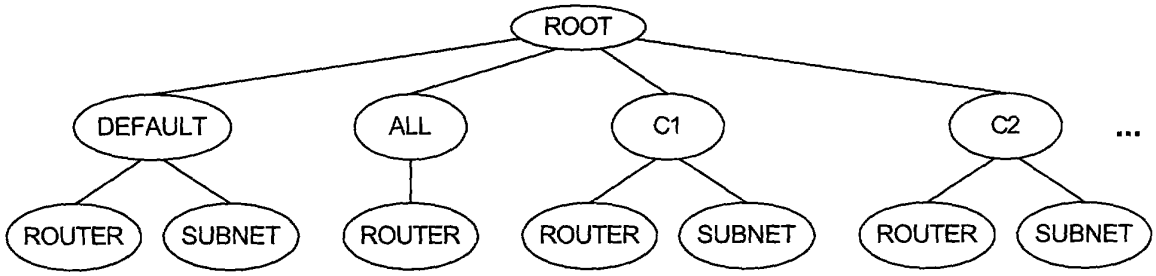
ROOT
DEFAULT
ALL
C1
C2
...
ROUTER
SUBNET
ROUTER
ROUTER
SUBNET
ROUTER
SUBNET

FIG. 5C

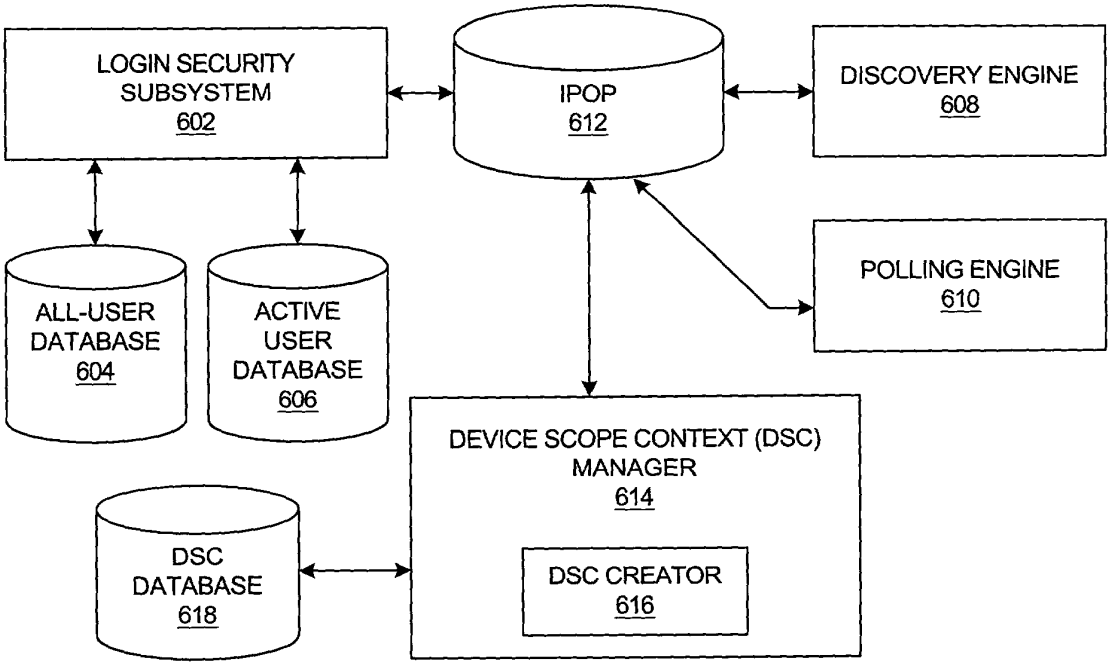
LOGIN SECURITY SUBSYSTEM 602
IPOP 612
DISCOVERY ENGINE 608
POLLING ENGINE 610
ALL-USER DATABASE 604
ACTIVE USER DATABASE 606
DEVICE SCOPE CONTEXT (DSC) MANAGER 614
DSC CREATOR 616
DSC DATABASE 618

FIG. 6

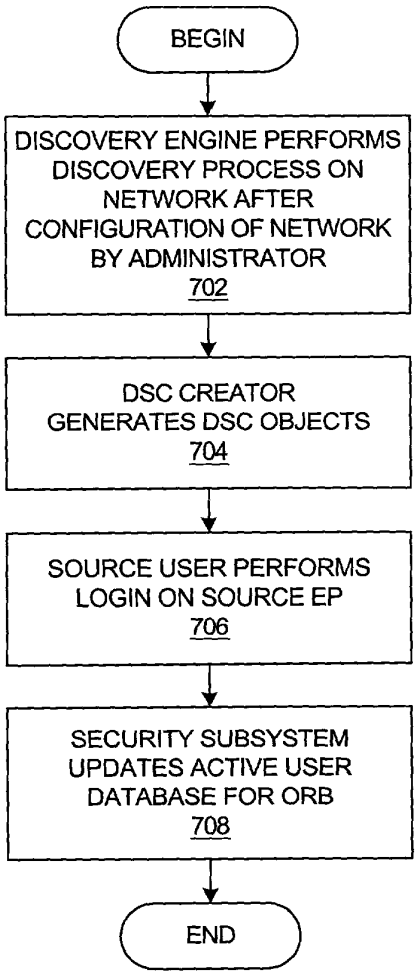
BEGIN
DISCOVERY ENGINE PERFORMS DISCOVERY PROCESS ON NETWORK AFTER CONFIGURATION OF NETWORK BY ADMINISTRATOR
702
DSC CREATOR GENERATES DSC OBJECTS
704
SOURCE USER PERFORMS LOGIN ON SOURCE EP
706
SECURITY SUBSYSTEM UPDATES ACTIVE USER DATABASE FOR ORB
708
END

FIG. 7A

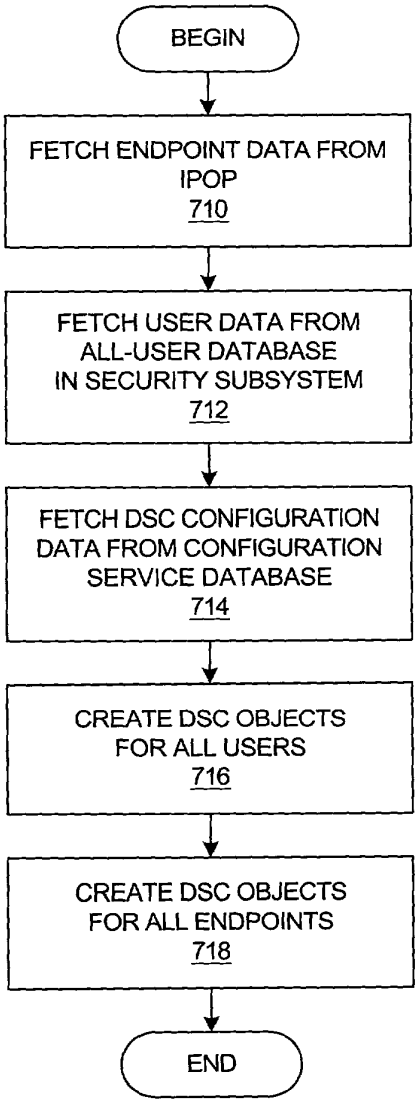
BEGIN
FETCH ENDPOINT DATA FROM IPOP
710
FETCH USER DATA FROM ALL-USER DATABASE IN SECURITY SUBSYSTEM
712
FETCH DSC CONFIGURATION DATA FROM CONFIGURATION SERVICE DATABASE
714
CREATE DSC OBJECTS FOR ALL USERS
716
CREATE DSC OBJECTS FOR ALL ENDPOINTS
718
END

FIG. 7B

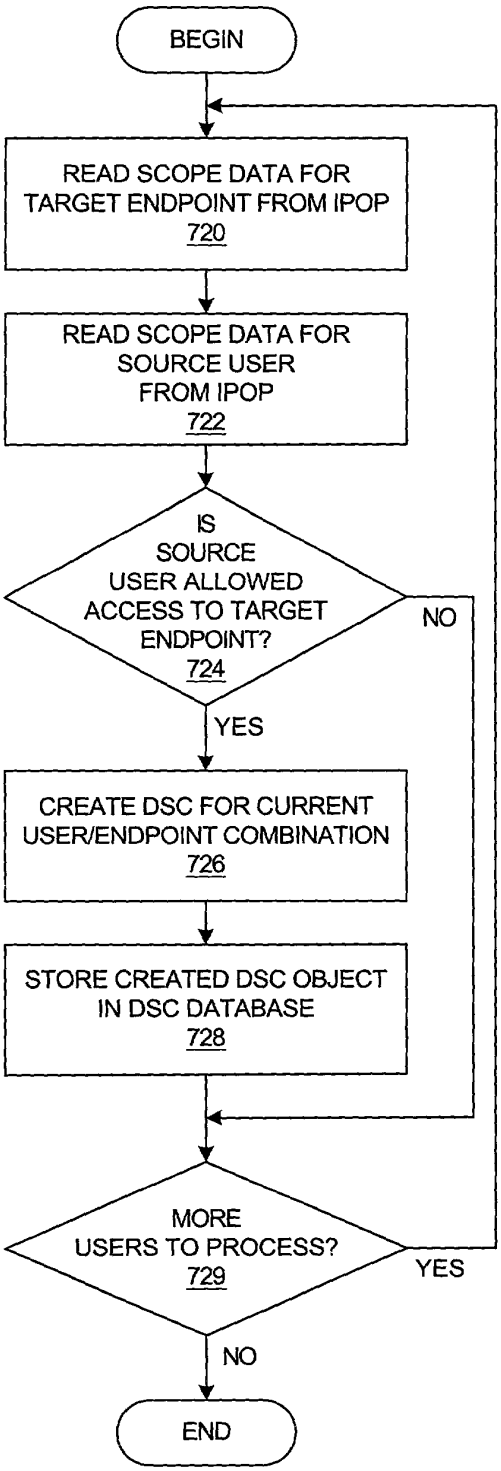
BEGIN
READ SCOPE DATA FOR TARGET ENDPOINT FROM IPOP 720
READ SCOPE DATA FOR SOURCE USER FROM IPOP 722
IS SOURCE USER ALLOWED ACCESS TO TARGET ENDPOINT? 724
NO
YES
CREATE DSC FOR CURRENT USER/ENDPOINT COMBINATION 726
STORE CREATED DSC OBJECT IN DSC DATABASE 728
MORE USERS TO PROCESS? 729
YES
NO
END

FIG. 7C

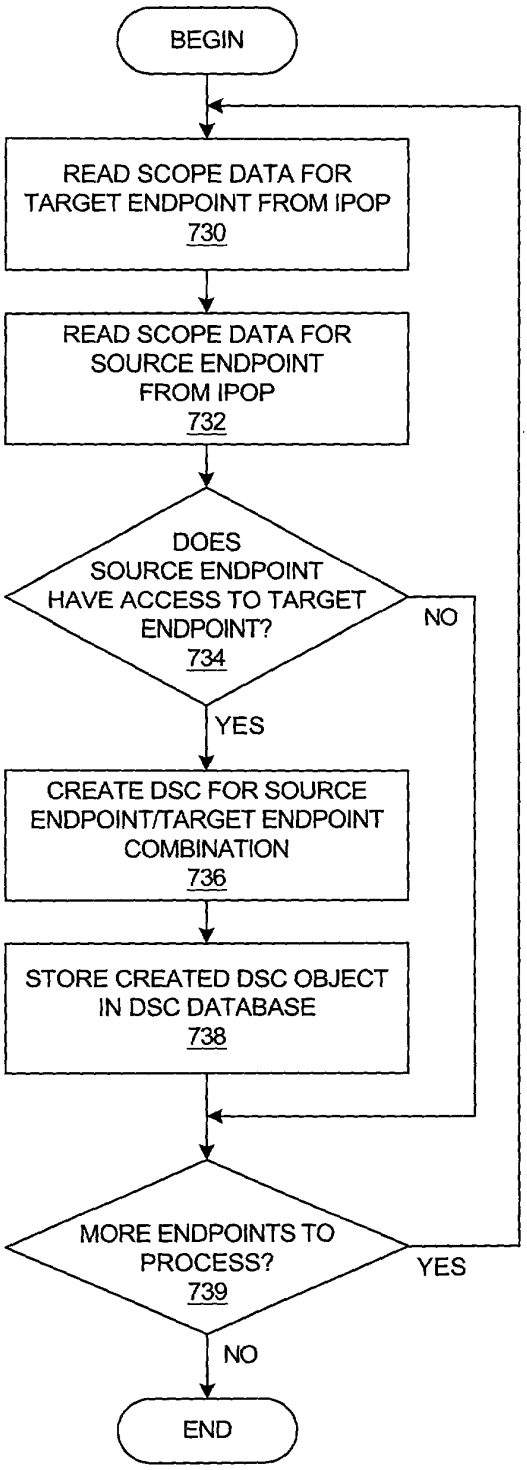
BEGIN
READ SCOPE DATA FOR TARGET ENDPOINT FROM IPOP 730
READ SCOPE DATA FOR SOURCE ENDPOINT FROM IPOP 732
DOES SOURCE ENDPOINT HAVE ACCESS TO TARGET ENDPOINT? 734
NO
YES
CREATE DSC FOR SOURCE ENDPOINT/TARGET ENDPOINT COMBINATION 736
STORE CREATED DSC OBJECT IN DSC DATABASE 738
MORE ENDPOINTS TO PROCESS? 739
YES
NO
END

FIG. 7D

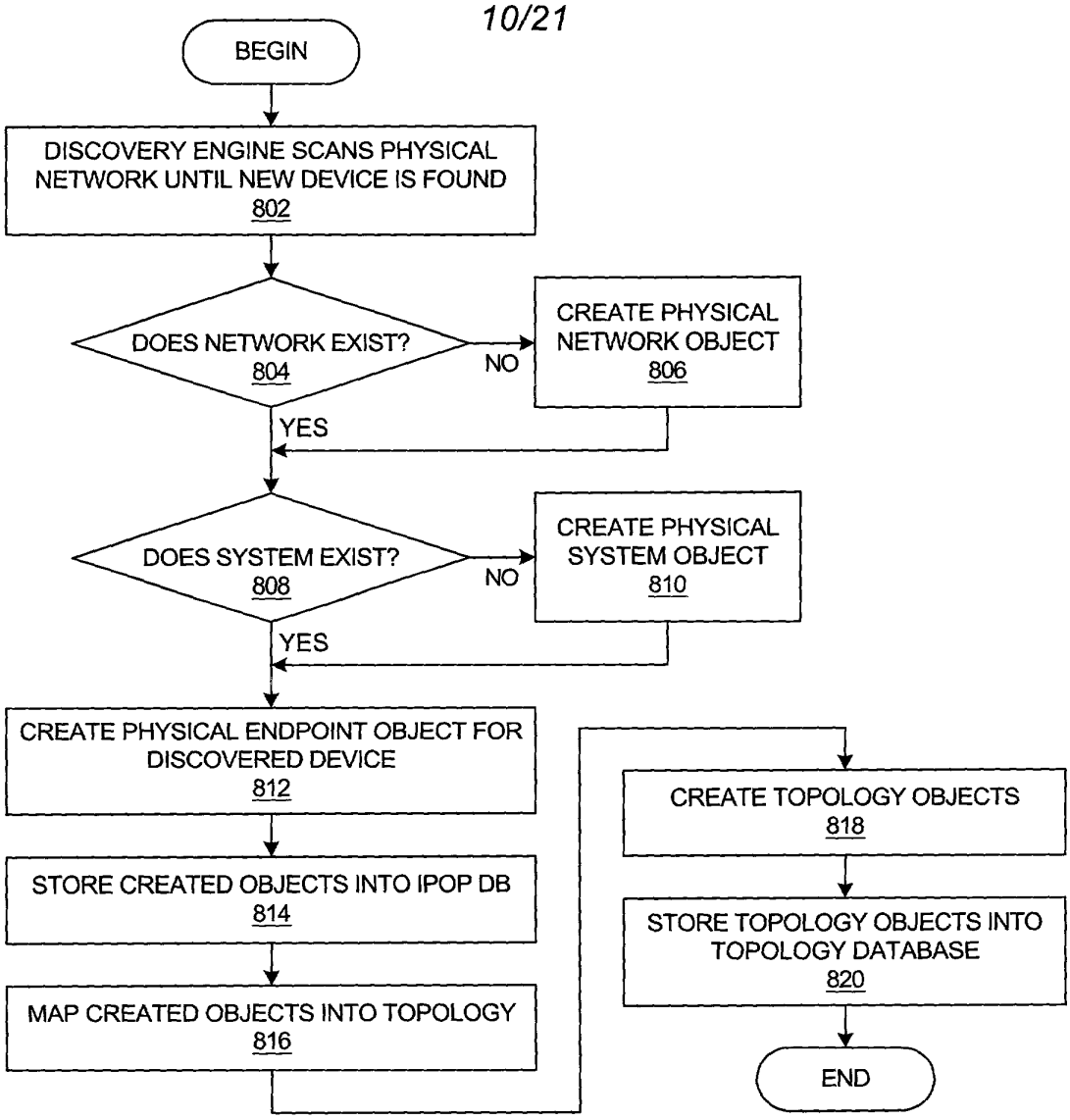
BEGIN
DISCOVERY ENGINE SCANS PHYSICAL NETWORK UNTIL NEW DEVICE IS FOUND
802
DOES NETWORK EXIST?
804
NO
CREATE PHYSICAL NETWORK OBJECT
806
YES
DOES SYSTEM EXIST?
808
NO
CREATE PHYSICAL SYSTEM OBJECT
810
YES
CREATE PHYSICAL ENDPOINT OBJECT FOR DISCOVERED DEVICE
812
STORE CREATED OBJECTS INTO IPOP DB
814
MAP CREATED OBJECTS INTO TOPOLOGY
816
CREATE TOPOLOGY OBJECTS
818
STORE TOPOLOGY OBJECTS INTO TOPOLOGY DATABASE
820
END

FIG. 8A

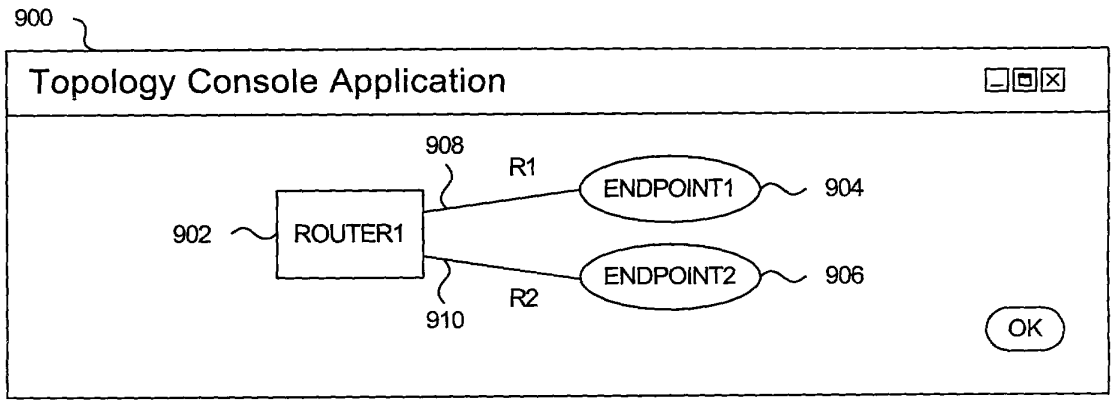
900
Topology Console Application
908
R1
ENDPOINT1
904
902
ROUTER1
ENDPOINT2
906
910
R2
OK

FIG. 9A

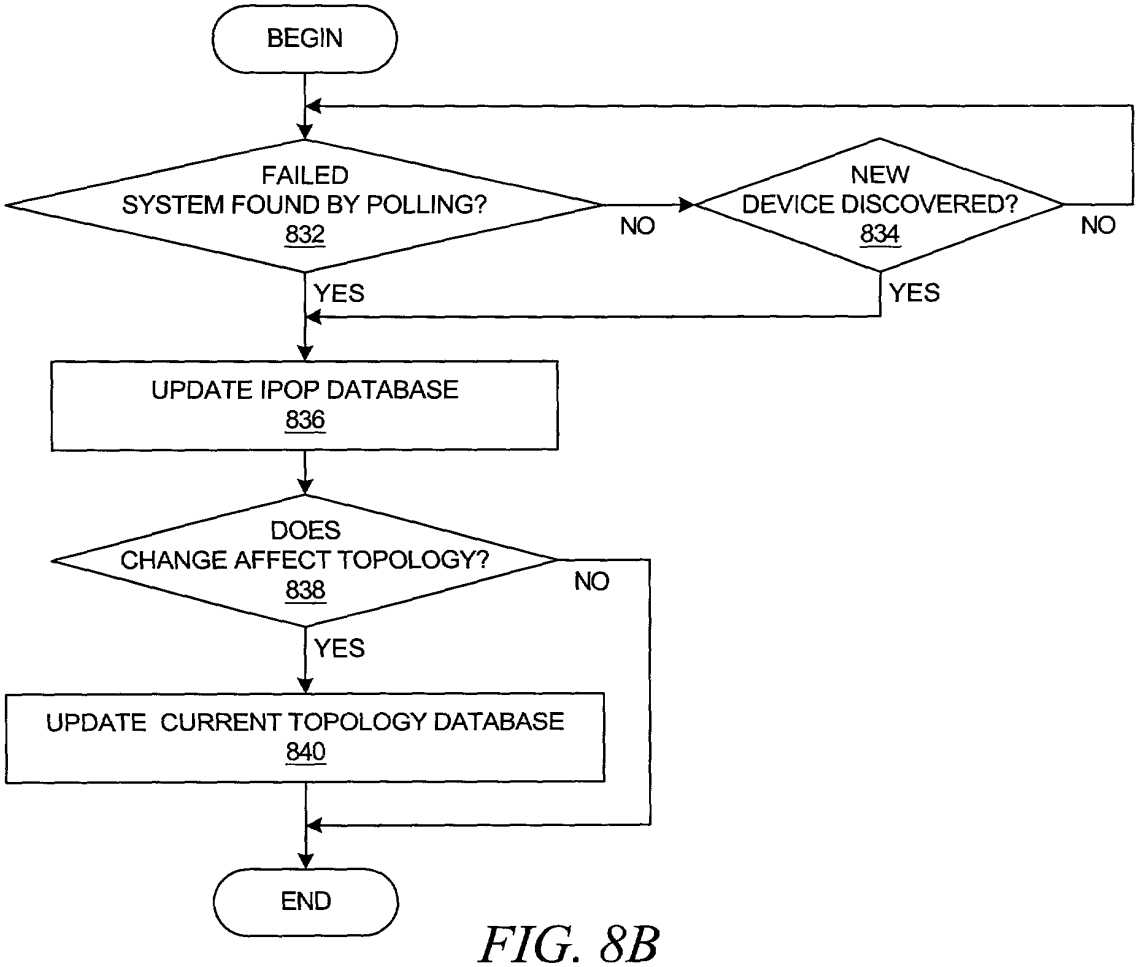
BEGIN
FAILED SYSTEM FOUND BY POLLING? 832
NO
NEW DEVICE DISCOVERED? 834
NO
YES
YES
UPDATE IPOP DATABASE 836
DOES CHANGE AFFECT TOPOLOGY? 838
NO
YES
UPDATE CURRENT TOPOLOGY DATABASE 840
END

FIG. 8B

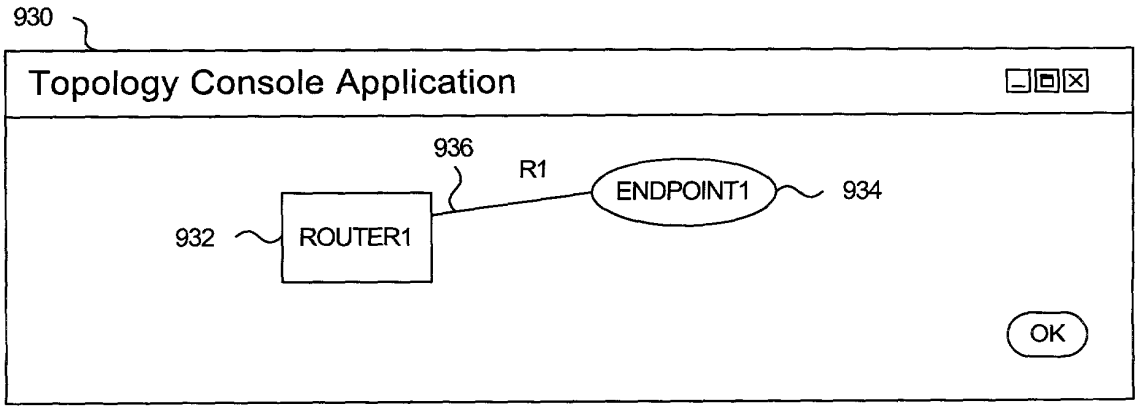
930
Topology Console Application
936
R1
ENDPOINT1
934
932
ROUTER1
OK

FIG. 9B

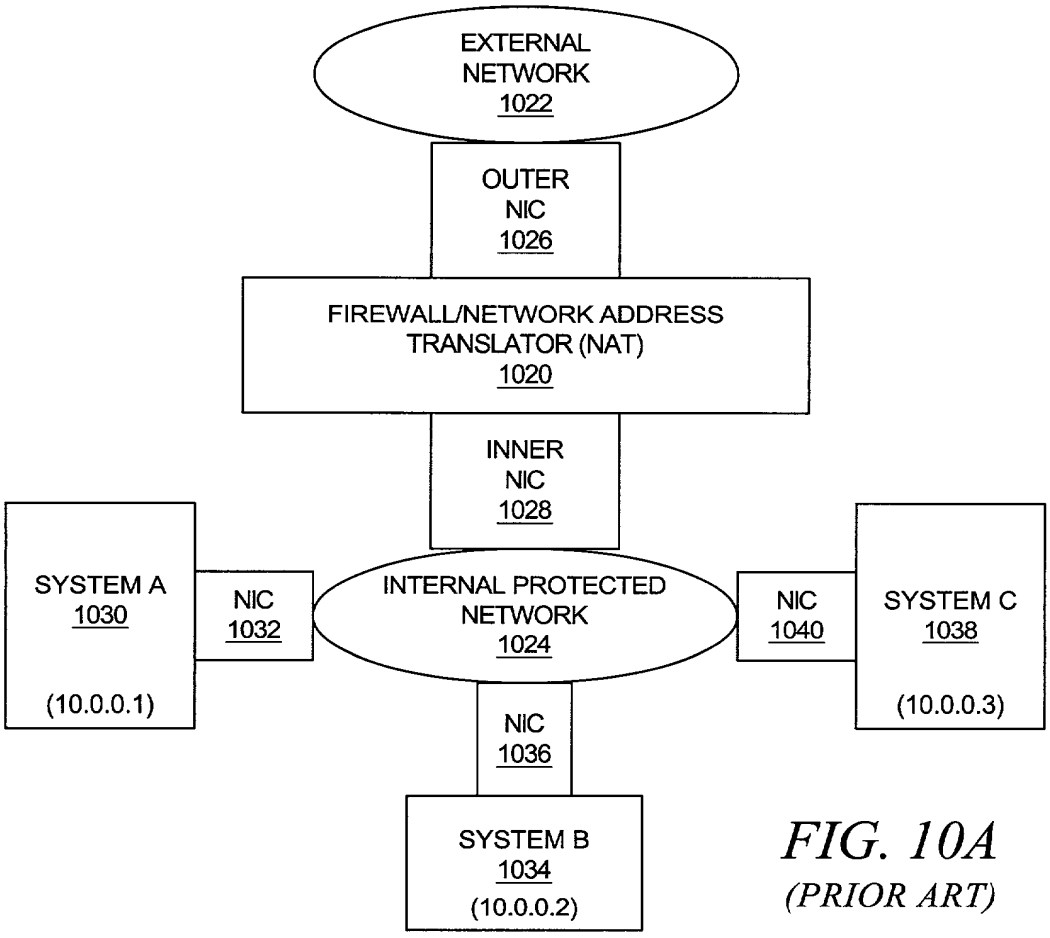

FIG. 10A
(PRIOR ART)

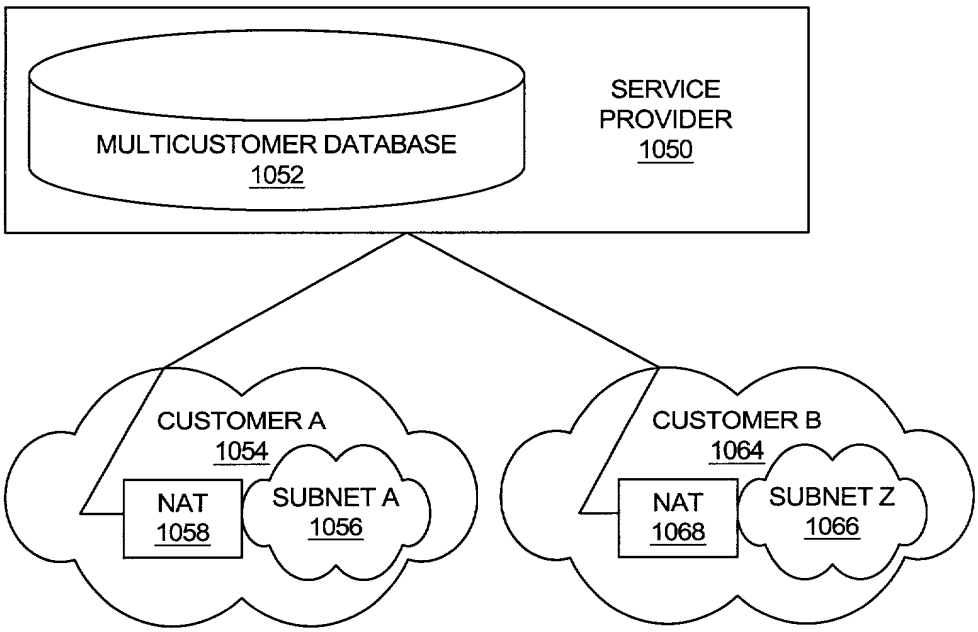

FIG. 10B

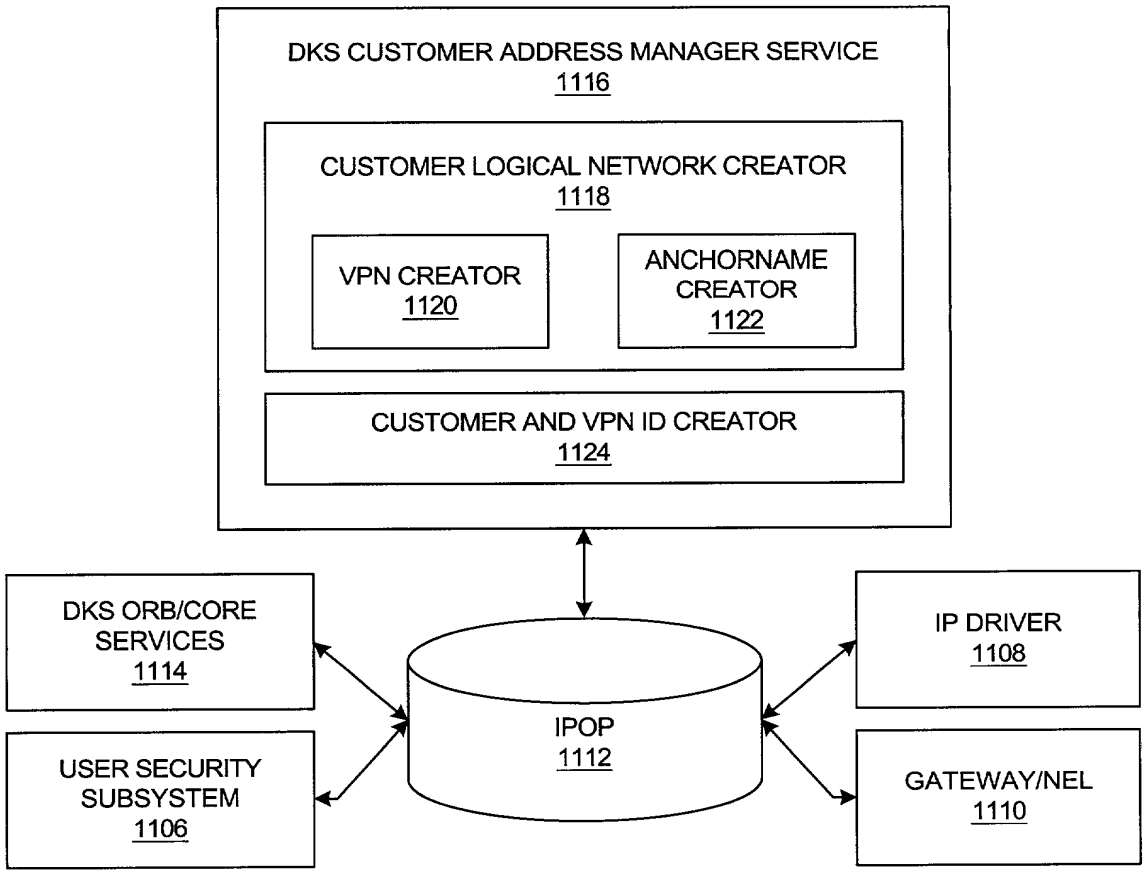

FIG. 11A

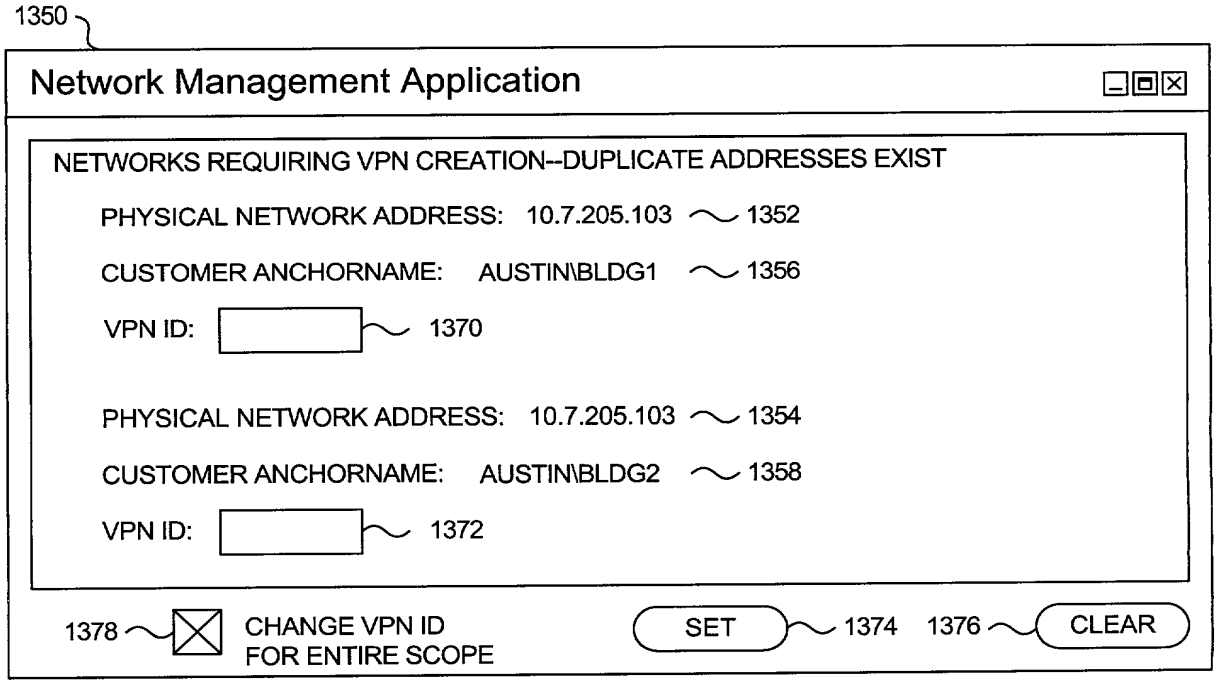

FIG. 13

```
Public Class IPActionObject {

        Endpoint sourceEP;
        Endpoint targetEP;

        // CONSTRUCTOR
        IPActionObject( Endpoint targetEP, Endpoint sourceEP ) {
        .
        .
        .
        }
        VOID performAction( ) // EXECUTES ACTION METHOD
.
.
.
}
```

FIG. 11B

```
Public Class Endpoint {

        // public variables
        long    EPObjectID;  // ID to object (both private and public network addresses)
        InetAddress   EPIPAddress;        // physical network address (private or public)
        long    EPVPN;       // virtual private network ID

        //get/set of variables
        public long             getObjectID( ) { ... }
        public InetAddress      getPAddress( ) { ... }
        public long             getVPN( ) { ... }

}
```

FIG. 11C

```
Public Class EndpointCustomer extends Endpoint {

        public getVPNGW( ) {
                //gets the only gateway which has access to a particular private network
        .
        .
        .
        }
        //private variables only set/accessed by EP creator IPOP
        long    customerHashNumber;
        String  customerName;
        String  customerAnchorPath;
        Long    objectIoFPrivateGatewayRoute

}
```

FIG. 11D

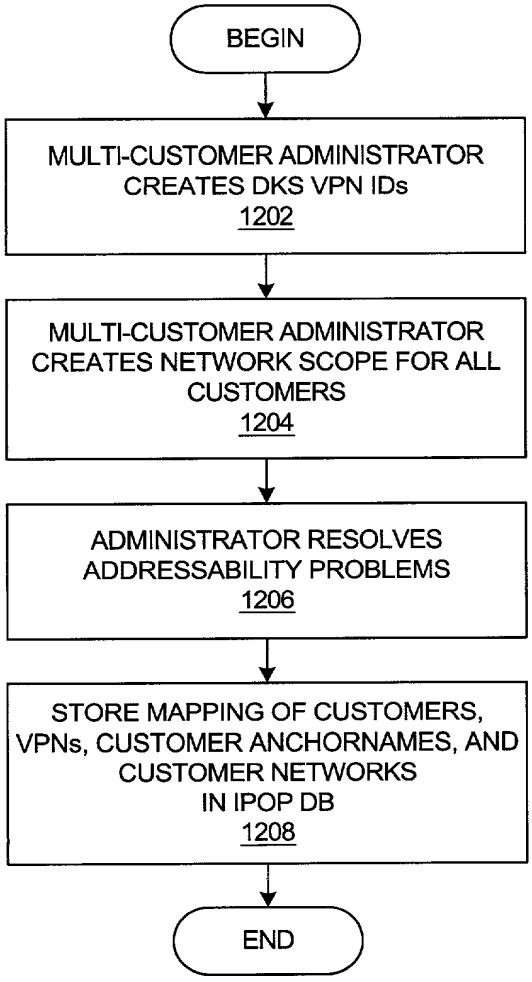

FIG. 12A

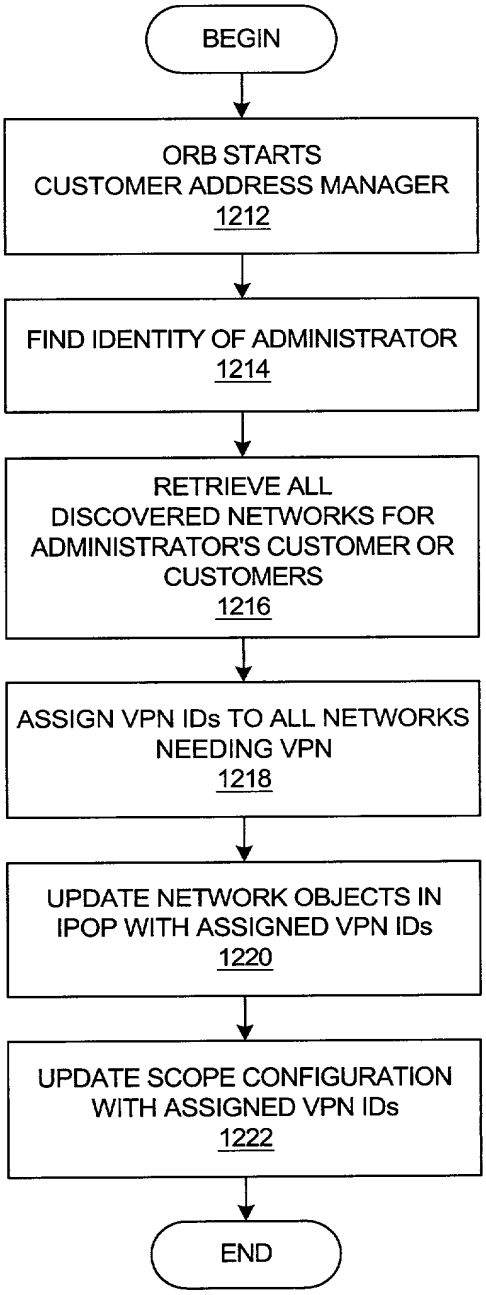

FIG. 12B

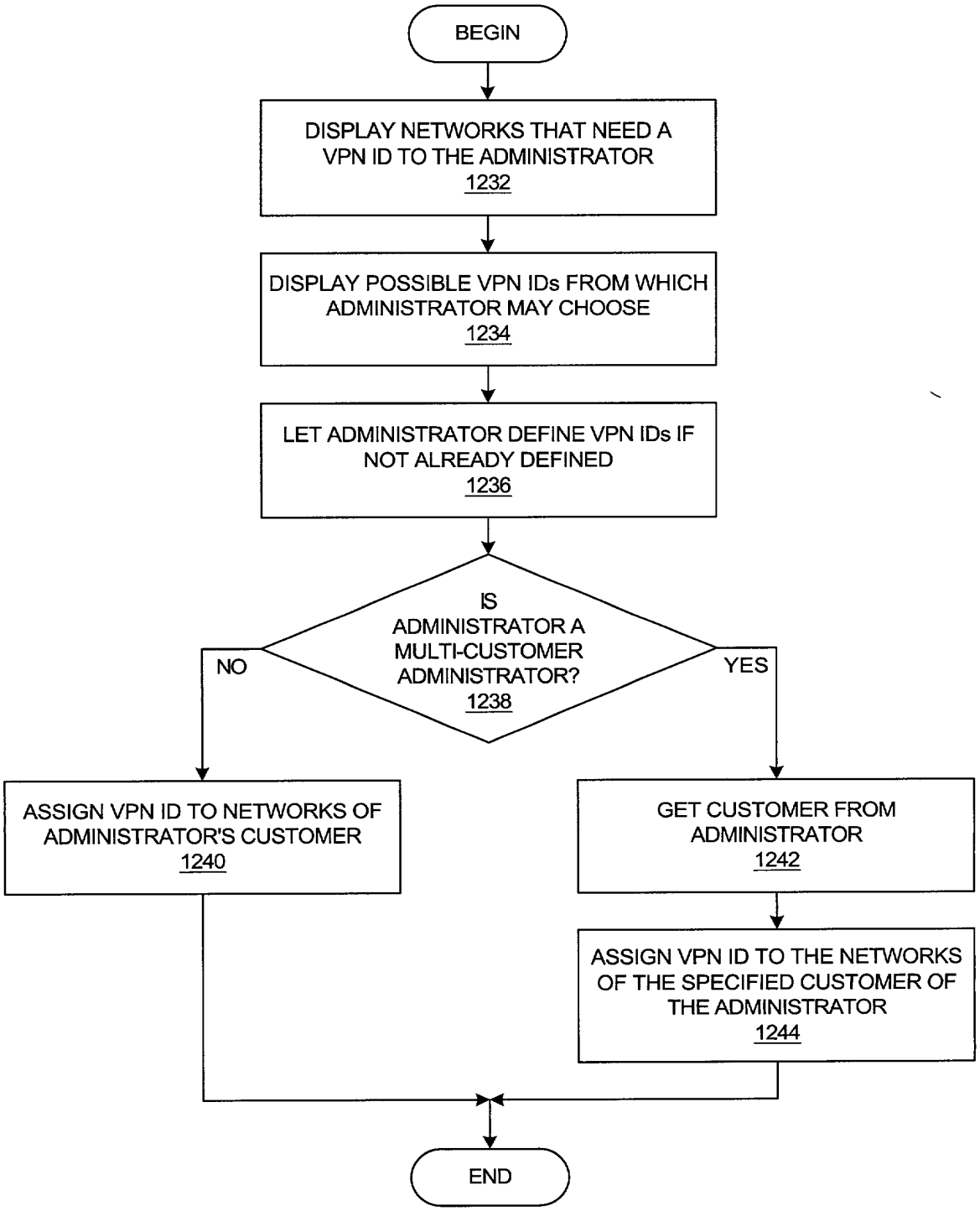
BEGIN
DISPLAY NETWORKS THAT NEED A VPN ID TO THE ADMINISTRATOR
1232
DISPLAY POSSIBLE VPN IDs FROM WHICH ADMINISTRATOR MAY CHOOSE
1234
LET ADMINISTRATOR DEFINE VPN IDs IF NOT ALREADY DEFINED
1236
IS ADMINISTRATOR A MULTI-CUSTOMER ADMINISTRATOR?
1238
NO
YES
ASSIGN VPN ID TO NETWORKS OF ADMINISTRATOR'S CUSTOMER
1240
GET CUSTOMER FROM ADMINISTRATOR
1242
ASSIGN VPN ID TO THE NETWORKS OF THE SPECIFIED CUSTOMER OF THE ADMINISTRATOR
1244
END

FIG. 12C

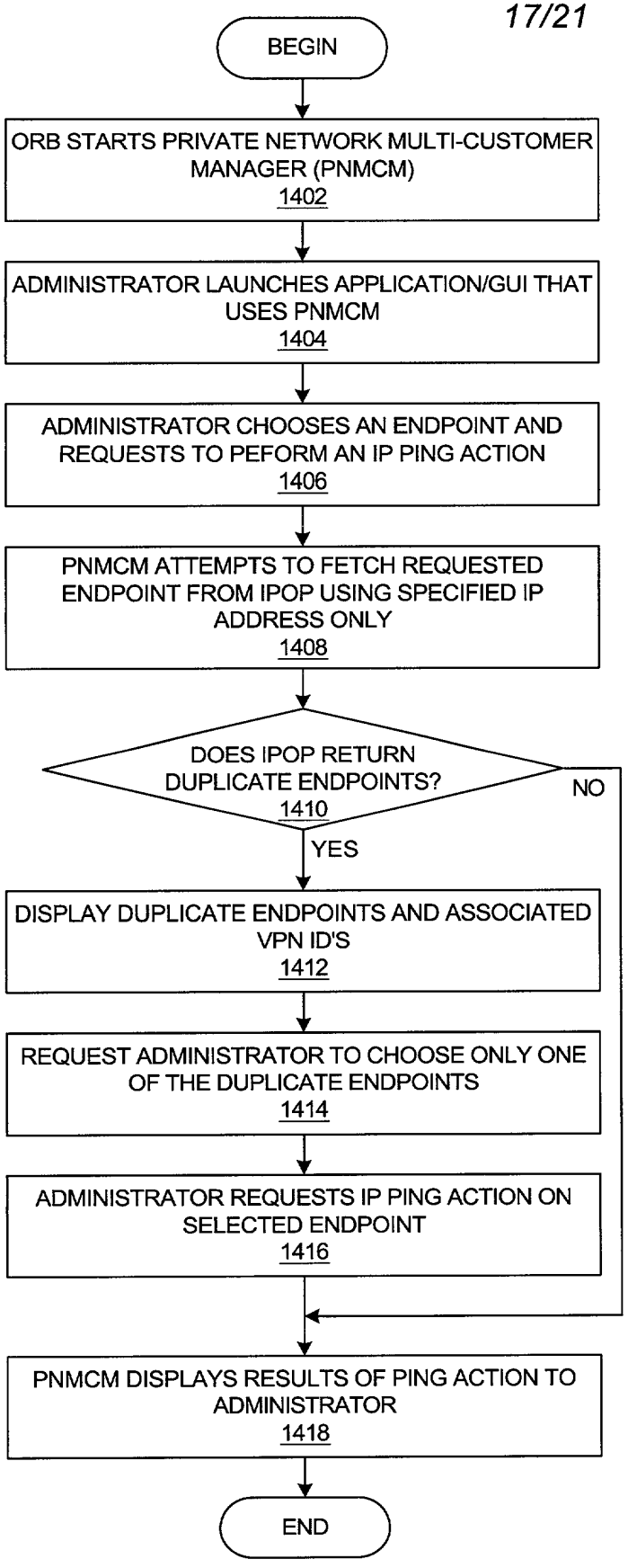

FIG. 14A

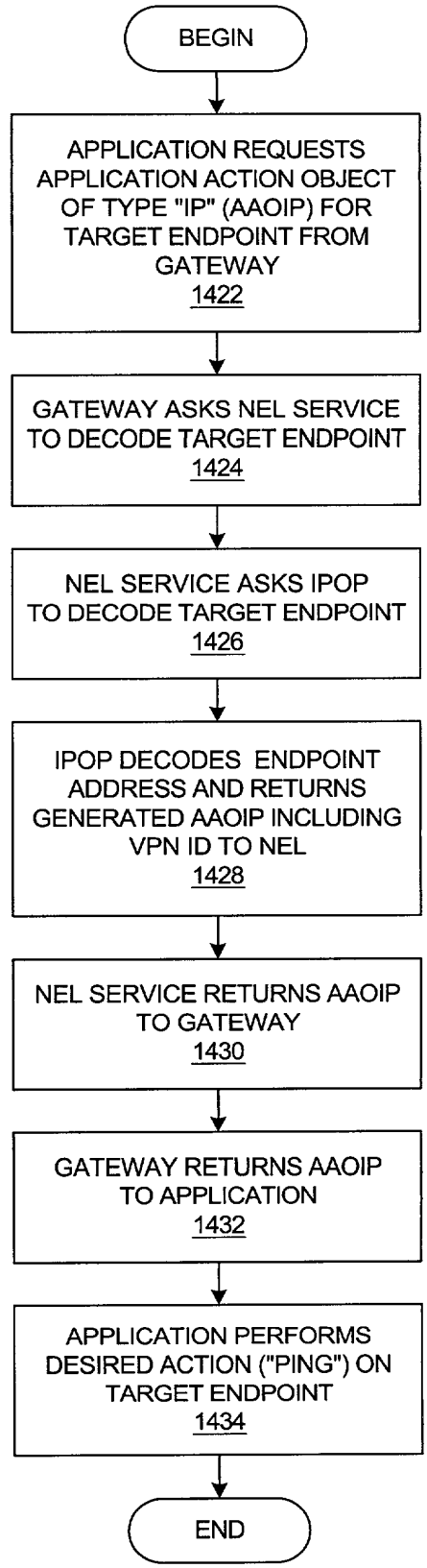

FIG. 14B

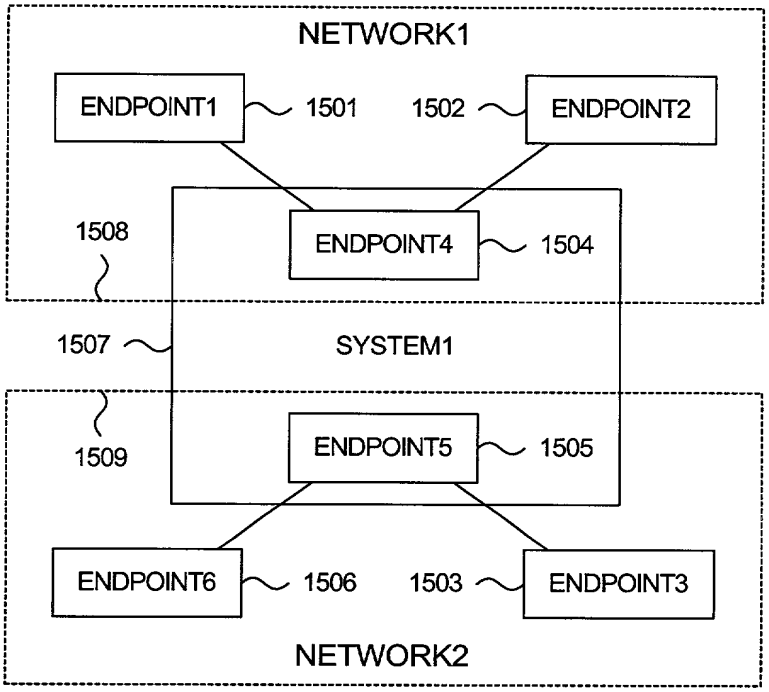
NETWORK1
ENDPOINT1
1501
1502
ENDPOINT2
1508
ENDPOINT4
1504
1507
SYSTEM1
1509
ENDPOINT5
1505
ENDPOINT6
1506
1503
ENDPOINT3
NETWORK2

FIG. 15

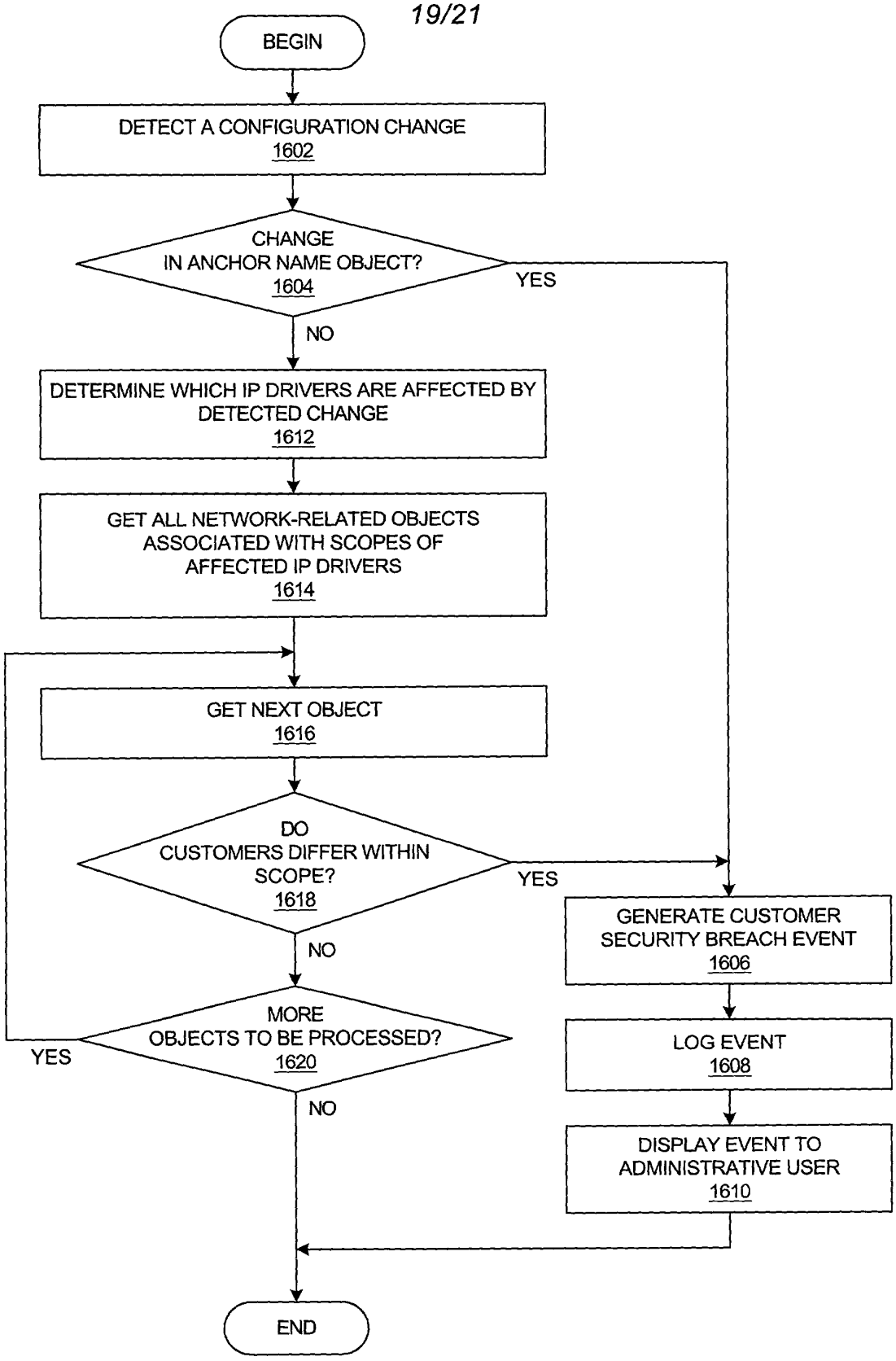
BEGIN
DETECT A CONFIGURATION CHANGE
1602
CHANGE IN ANCHOR NAME OBJECT?
1604
YES
NO
DETERMINE WHICH IP DRIVERS ARE AFFECTED BY DETECTED CHANGE
1612
GET ALL NETWORK-RELATED OBJECTS ASSOCIATED WITH SCOPES OF AFFECTED IP DRIVERS
1614
GET NEXT OBJECT
1616
DO CUSTOMERS DIFFER WITHIN SCOPE?
1618
YES
NO
GENERATE CUSTOMER SECURITY BREACH EVENT
1606
LOG EVENT
1608
DISPLAY EVENT TO ADMINISTRATIVE USER
1610
MORE OBJECTS TO BE PROCESSED?
1620
YES
NO
END

FIG. 16

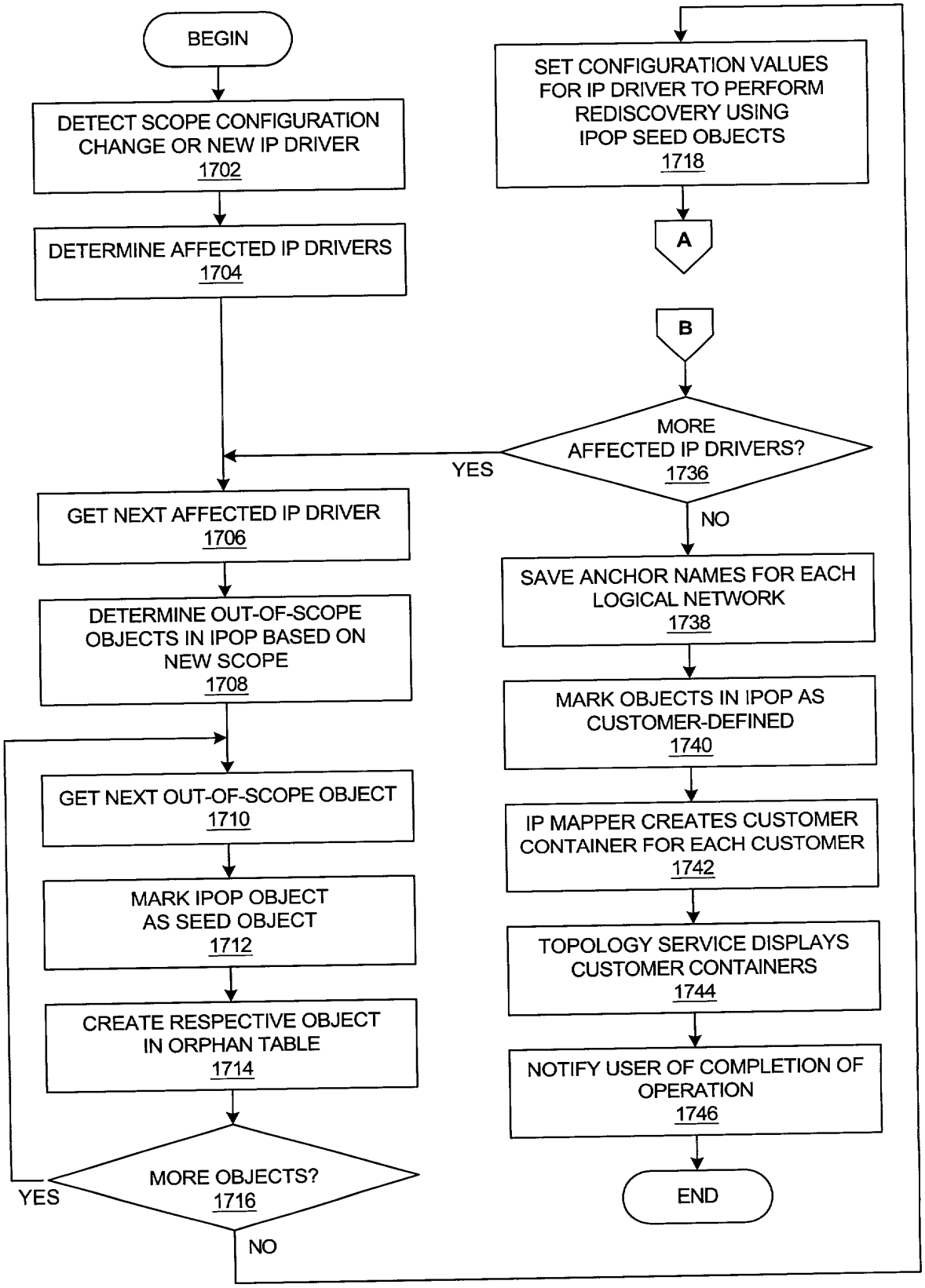
BEGIN
DETECT SCOPE CONFIGURATION CHANGE OR NEW IP DRIVER 1702
DETERMINE AFFECTED IP DRIVERS 1704
GET NEXT AFFECTED IP DRIVER 1706
DETERMINE OUT-OF-SCOPE OBJECTS IN IPOP BASED ON NEW SCOPE 1708
GET NEXT OUT-OF-SCOPE OBJECT 1710
MARK IPOP OBJECT AS SEED OBJECT 1712
CREATE RESPECTIVE OBJECT IN ORPHAN TABLE 1714
MORE OBJECTS? 1716
YES
NO
SET CONFIGURATION VALUES FOR IP DRIVER TO PERFORM REDISCOVERY USING IPOP SEED OBJECTS 1718
A
B
MORE AFFECTED IP DRIVERS? 1736
YES
NO
SAVE ANCHOR NAMES FOR EACH LOGICAL NETWORK 1738
MARK OBJECTS IN IPOP AS CUSTOMER-DEFINED 1740
IP MAPPER CREATES CUSTOMER CONTAINER FOR EACH CUSTOMER 1742
TOPOLOGY SERVICE DISPLAYS CUSTOMER CONTAINERS 1744
NOTIFY USER OF COMPLETION OF OPERATION 1746
END

FIG. 17A

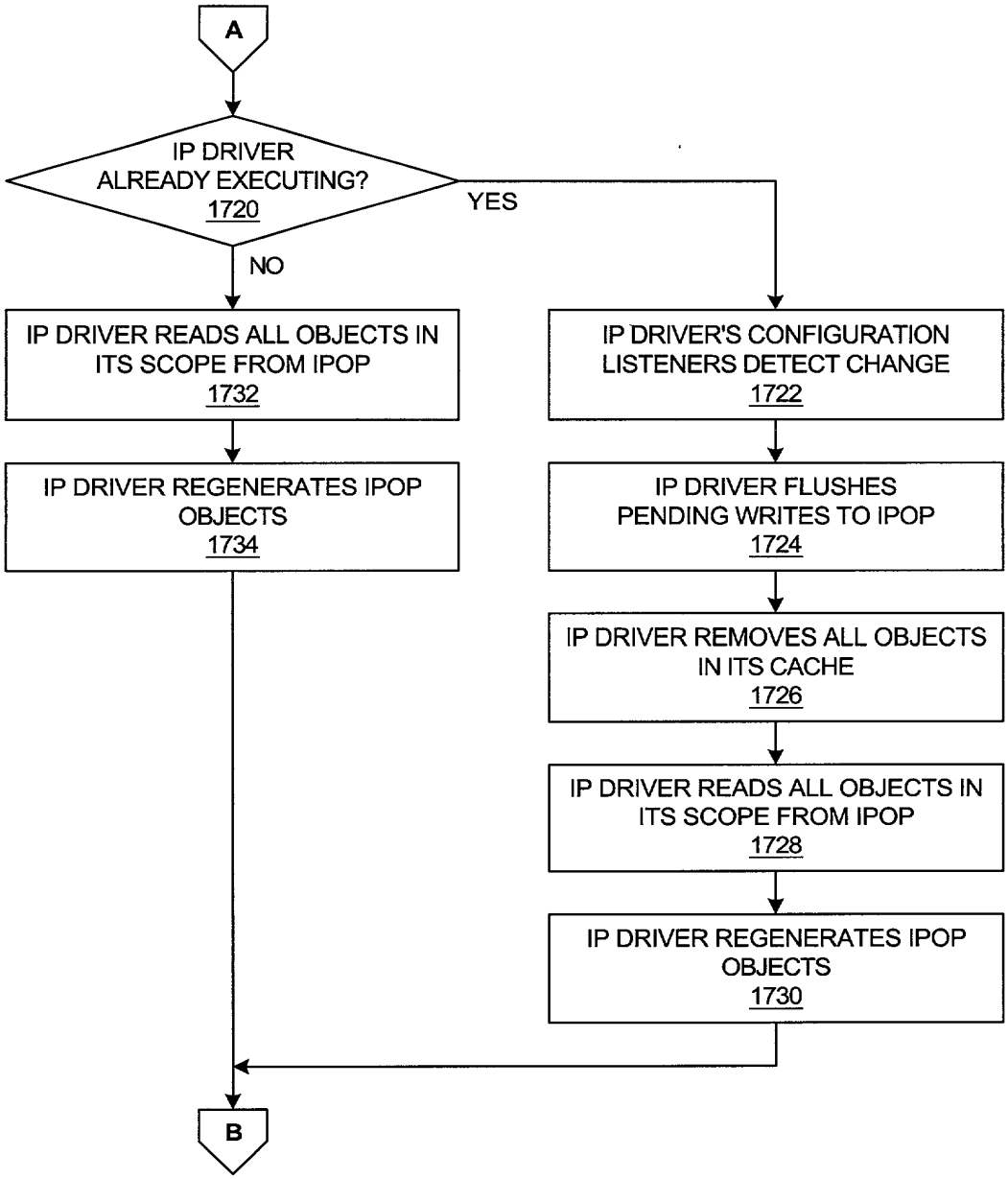
A
IP DRIVER ALREADY EXECUTING? 1720
YES
NO
IP DRIVER READS ALL OBJECTS IN ITS SCOPE FROM IPOP 1732
IP DRIVER REGENERATES IPOP OBJECTS 1734
IP DRIVER'S CONFIGURATION LISTENERS DETECT CHANGE 1722
IP DRIVER FLUSHES PENDING WRITES TO IPOP 1724
IP DRIVER REMOVES ALL OBJECTS IN ITS CACHE 1726
IP DRIVER READS ALL OBJECTS IN ITS SCOPE FROM IPOP 1728
IP DRIVER REGENERATES IPOP OBJECTS 1730
B

FIG. 17B